MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 3, 2016/2017

TCP 1211 – LOGIC PROGRAMMING

(All sections / Groups)

26th May 2017
9:00 am – 11:00 am
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 4 pages only excluding the cover page.
2. Attempt **FOUR** out of **FIVE** questions. .All questions carry equal marks and the distribution of the marks for each question is given.
2. Please write your answers in the answer booklet provided. Please write the question number of each answer clearly.

QUESTION (1)

(a) Define the relationships: `daughter, brother, aunt, nephew, cousin` and `grand-daughter` using the following Prolog clauses. [12 marks]

```
female(ann).
female(mary).
male(tom).
male(john).
parent(ann, john).
parent(ann, mary).
parent(tom, john).
parent(tom, mary).
sibling(X, Y):- parent(Z, X), parent(Z, Y), X \== Y.
```

(b) Draw a hierarchy to show the classification of syntactic data objects in Prolog. [3 marks]

QUESTION (2)

(a) Considering the following Prolog program:

```
dispList([]).
dispList([H|T]):- dispList(T),write('N'),write(H).
```

What would be the output of the following goals? [2 marks]

```
?- dispList([3,w,8,z]).
```

(b) Define a recursive relation `evenlist(L)`, which succeeds only for lists of even length. [2 marks]

(c) Define the predicate `sumlist(List,S)` so that Sum is the sum of a given list of numbers. For example: [3 marks]

```
?- sumlist([0,1,2,3],S).
   S=6
```

(d) For a certain purpose, a list of numbers is passable if it does not contain the number 5. Write Prolog code to check that a list is passable. [4 marks]

NOTE: Do not use predicate **"not"** or inequality. Do use **"cut"**.

(e) Define the relation `insert(List, E1, E2, Newlist)` which is true when `Newlist` consists of `List` with the element `E1` inserted next to element `E2`. [4 marks]

Example:

```
?- insert([a,b,c,d,e], d,f, Y).
   Y = [a, b, c, f, e];
```

Continue...

QUESTION (3)

(a) Given the following program:

```
student_of(peter, ann).
student_of(paul,  peter).
student_of(john, peter).
student_of(tom, ann).
friendly(maria).
friendly(paul).
friendly(ann).
friendly(peter).
friendly(tom).
likes(X, Y) :- student_of(Y, X),friendly(Y).
likes(tom,X) :- friendly(X),!.
invites(X, Y) :- student_of(Y, X),!,friendly(Y).
```

List all Prolog's answers to the following queries: [6 marks]

***Note:** for every extra answer you may write, 0.5 mark will be deducted.*

```
1. ?- invites (X, Y).
2. ?- student_of (Y,X) , not (friendly (Y) ).
3. ?- student_of(Y,X) , not(invites(X,Y) ).
4. ?- student_of (Y, X) , ! , friendly (X).
5. ?- likes(X, Y) , likes(Y, Z), !, not (invites(X, Z)).
6. ?- invites(X, Y) , not(invites(Y, X)).
```

(b) Projects in a company are described as a database with facts as follow:

```
% project(ProjectName,Manager,[Participants],StartTime,StopTime)
% ====================== Database ======================
          project(a,lisa,[],20,25).
          project(b,lisa,[erik],23,35).
          project(c,johnny,[],40,45).
          project(d,johnny,[lisa,bengt],47,55).
          project(e,johnny m1,[lisa],52,60).
          project(f,erik,[lisa,johnny],70,85).
          project(g,bengt,[lisa,johnny],90,95).
```

`StartTime` and `StopTime` are integer numbers, `Projectmame` and `Manager` are constants and `Participants` is a list of unique constants. For the above database, define Prolog rules that can answer the following queries: [6 marks]

1. Find two projects `PI, P2` with the same manager.
2. Find two projects `PI, P2` with the same participant(s).
3. Find two projects `PI, P2` with overlapping time.

(c) Rewrite the following formulas in ***prefix*** format (order).

```
1. Y + X * Z
2. A is B + C
```

[3 marks]

QUESTION (4)

(a) Given the following Prolog program, draw the SLD tree for the query `?- p(X).`

[5 marks]

```
m(c).                    %1
r(b).                    %2
s(a).                    %3
s(b).                    %4
u(c).                    %5
p(X):-u(X).              %6
p(X):-q(X),r(X).         %7
p(X):-m(X),n(X).         %8
q(X):-s(X).              %9
```

(b) What answer do you expect for each of the following goals? [6 marks]

```
1. ?- A=:=B.
2. ?- 14 = 2*7.
3. ?- abc = X, X == abc.
4. ?- 10+5 == 5+10.
5. ?- X=3,Y is X+2-2,X=:=Y.
6. ?- .(.(a,[]),.(.(b,[]),.(.(c,[]),[]))) = X.
```

(c) If E and F are two simple expressions, define formally the most general unifier ***mgu*** of E and F , find the ***mgu*** of the terms: [4 marks]

`f(g(X),h(Y),X,Y)` and `f(Z,W,k(W,a),b)`

QUESTION (5)

(a) What is negation as failure? [3 marks]

(b) Discuss the common misuse of negation as failure. [3 marks]

(c) SLD-resolution procedure is both sound and complete.

1) In what conditions a reasoning system is claimed to be sound?
2) In what conditions a reasoning system is claimed to be complete?

[3 marks]

(d) The state-transition concept is the basic idea behind the problem-solving techniques using search. In order to solve a problem using search, two main tasks are required. What are these tasks? [3 marks]

(e) Give TWO differences between depth-first search and breadth-first search. [3 marks]

End of page.